Sub-pixel Estimation of Local Extrema

Donald G. Bailey
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
D.G.Bailey@massey.ac.nz

Abstract

When estimating the location of features such as lines and edges to sub-pixel accuracy, it is necessary to have a model of the feature to fit the available data to. This paper considers detecting local minima and maxima, and proposes several models for estimating the sub-pixel location of the corresponding features. These methods are compared by detecting the locations of lines of various widths, and edges. The effects of lens blur, quantisation, and noise are also considered. The centre of gravity is effective at locating narrow lines, while a parabolic fit is the best for thicker lines and edges. For typical image contrasts and signal to noise ratios, a sub-pixel accuracy of 1 to 2 % of the width of a pixel can be expected.

Keywords: line detection, edge detection, models, centre of gravity, feature location, accuracy

1 Introduction

Many applications require, or can benefit from, locating features to sub-pixel accuracy. Examples are:

- Gauging: locating edges of objects to measure size for inspection.
- Camera calibration: locating feature points (lines, spots, edges) (eg [1-3]).
- Registration: determining offsets between objects for template matching, or calculating disparity in stereo imaging (eg [4]).
- Compression: determining offset for motion compensation.

Locating any feature to sub-pixel accuracy requires first of all that the feature can be reliably detected. This implies that the feature is sufficiently far (usually at least 2 pixels) from other similar features so that it can be resolved unambiguously. A second requirement is that the feature can be modelled with only a few parameters. By fitting the available data (pixel values) to the model, the location of the feature may be determined with accuracy significantly better than 1 pixel.

Correlation with a template results in a set of local maxima, the locations of correspond to locations of the template within the image. Alternatively, if minimum absolute difference is used, a set of local minima results [5].

Lines less than 2 pixels thick may be detected directly by looking perpendicular to the line. The line position corresponds to either the minima or maxima depending on whether it is a dark line against a light background, or vice versa. Edges may be converted into lines by an edge detection filter.

While it is acknowledged that higher order models may be used for locating lines and edges to sub-pixel accuracy, this paper is concerned with locating each pixel on the line or edge prior to fitting a higher order model. The approach here estimates the sub-pixel position of the line or edge by considering each candidate pixel and its two neighbours, either horizontally or vertically (depending on whether the line is vertical or horizontal respectively). This is repeated for each potential candidate pixel on the line or edge. The place of higher order models is then to link the adjacent detected pixels, and to any detection noise. The sub-pixel detection of each candidate point will reduce the systematic error that the higher order models have to contend with.

2 Models

All of the models discussed in this paper use the extreme pixel value along with the pixel values on either side of it to estimate the position of the extremum to sub-pixel accuracy. If looking at narrow features such as lines, and detected edges, it does not make sense to go any wider than this because the data outside this width is usually of little value.

The notation used throughout this paper is illustrated in figure 1. A pixel's position is taken at the centre of that pixel. Pixels are numbered relative to the local extreme, which has a value of p_0. The pixel at 1 has value p_+, and that at –1 has value p_-. The true extremum is located at a distance x from the extreme pixel value, where $-0.5 < x < 0.5$.

Using just these three pixel values, the different models give a value for x.

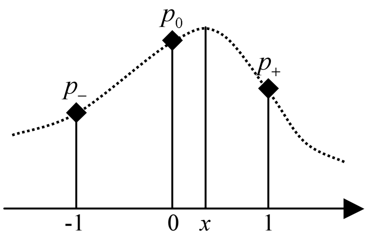

Figure 1: Local maximum and pixel definitions.

2.1 Parabola

Given that the true curve is smooth, a common approach is to approximate it in the vicinity of the extreme point by a parabola.

$$y = ax^2 + bx + c \tag{1}$$

This parameterised curve is then solved to find the true position of the local extremum by differentiating and setting the derivative to zero.

$$\frac{dy}{dx} = 2ax + b = 0 \qquad \therefore x = \frac{-b}{2a} \tag{2}$$

Substituting the pixel values for each of the 3 known data points into equation (1) gives:

$$\begin{aligned} p_- &= a - b + c \\ p_0 &= c \\ p_+ &= a + b + c \end{aligned} \tag{3}$$

which when solved and substituted into equation (2) results in the estimate of the location of the true extreme point:

$$x = \frac{p_+ - p_-}{4p_0 - 2(p_+ + p_-)} \tag{4}$$

In this analysis, no assumption is made as to whether the centre point is a local minimum or maximum. Therefore this equation may be used for either.

2.2 Pyramid

In many situations, however, it is known that the peak is not parabolic in shape. For example if the peak results from the correlation of an object with a template, then generally the peak will be shaped like a pyramid [4]. This is particularly the case if correlating an object with approximately uniform intensity with a binary template.

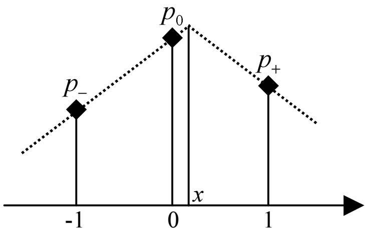

Figure 2: Pyramid interpolation of local maximum.

To locate the peak of the pyramid, it is necessary to assume that the slope is constant over at least the maximum and an adjacent pixel. If $p_+ > p_-$ as shown in figure 2, then the slope on the left may be determined from p_0 and p_-. The slope on the right is negative this, so the equation for the line on the right can be derived from the slope and the single point. The two lines are:

$$\begin{array}{ll} \text{Left} & y = (p_0 - p_-)x + p_0 \\ \text{Right} & y = (p_- - p_0)x + (p_+ + p_0 - p_-) \end{array} \tag{5}$$

Solving for where these intersect gives

$$x = \frac{p_+ - p_-}{2(p_0 - p_-)} \tag{6}$$

Similarly, if $p_- > p_+$ then the slope may be determined from p_0 and p_+, and a line of opposite slope fitted through p_-. This gives the peak of the pyramid at

$$x = \frac{p_+ - p_-}{2(p_0 - p_+)} \tag{7}$$

Combining these equations (6) and (7) gives

$$x = \frac{p_+ - p_-}{2(p_0 - \min(p_-, p_+))} \tag{8}$$

The analysis here assumed that p_0 was a local maximum. If instead it was a local minimum (for example template matching using minimum absolute difference), then equation (8) becomes

$$x = \frac{p_+ - p_-}{2(p_0 - \max(p_-, p_+))} \tag{9}$$

2.3 Centre of Gravity

When detecting a line, or locating an edge after an edge detection filter, the peak is relatively narrow. If the background is zero then the location of the peak may be found from the centre of gravity:

$$x = \frac{\sum x_i p_i}{\sum p_i} = \frac{p_+ - p_-}{p_0 + p_+ + p_-} \tag{10}$$

If the background is not equal to zero, or if looking for a local minimum (where the background is not zero) then the background must be estimated and subtracted from each of the values. Equation (10) then becomes

$$x = \frac{p_+ - p_-}{p_0 + p_+ + p_- - 3p_{bg}} \tag{11}$$

where p_{bg} is the background level.

If the line is sufficiently narrow, that one of the three pixels will be the background, then for detecting local maxima this becomes:

$$x = \frac{p_+ - p_-}{p_0 + p_+ + p_- - 3\min(p_+, p_-)} \tag{12}$$

or for local minima:

$$x = \frac{p_+ - p_-}{p_0 + p_+ + p_- - 3\max(p_+, p_-)} \tag{13}$$

2.4 Rectangular Section

This model is appropriate for looking at lines that are wider than 1 pixel. It assumes that the centre pixel is completely covered by the line, and that the side pixels partly overlap the line as shown in figure 3. Their pixel values are given from the proportion of the pixel covered by the edge.

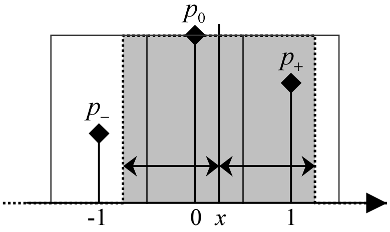

Figure 3: Line overlapping pixels. Pixel values p_+ and p_- are determined from extent of overlap with line.

Since the centre pixel is completely occupied by the line, it can be used to give the intensity of the line. From this, the edges of the line are given by:

$$\begin{aligned} x_+ &= p_+/p_0 \\ x_- &= p_-/p_0 \end{aligned} \tag{14}$$

The centre of the line is therefore given from the centre of the two edges:

$$x = \frac{(x_+ + \frac{1}{2}) - (x_- + \frac{1}{2})}{2} = \frac{p_+ - p_-}{2p_0} \tag{15}$$

Again, the assumption here is that the background is zero. If the background is not zero (or if locating a dark line on a light background) then equation (15) is modified to:

$$x = \frac{p_+ - p_-}{2(p_0 - p_{bg})} \tag{16}$$

This reduces to equation (8) or (9) if one of the three pixels is the background.

If the line is exactly 1 pixel wide, as shown in figure 4, then the model is simplified. Consider first a light line on a dark background with the $p_+ > p_-$. The pixel values are given by:

$$\begin{aligned} p_- &= p_{bg} \\ p_0 &= p_{bg} + a(1-x) \\ p_+ &= p_{bg} + ax \end{aligned} \tag{17}$$

Eliminating a and p_{bg} and solving this for x gives

$$x = \frac{p_+ - p_-}{p_0 + p_+ - 2p_-} \tag{18}$$

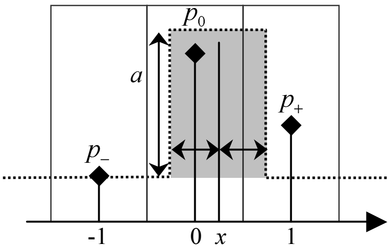

Figure 4: A 1 pixel wide line.

If we also consider the case when $p_- > p_+$ then this reduces to the centre of gravity in equation (12).

2.5 Sobel Filtered Edge

When locating edges to sub-pixel accuracy, it is important that the edge-detecting filter does not introduce a shift. Therefore filters, such as the Sobel filter [6], which take a balanced difference are appropriate. Figure 5 illustrates the effect of applying a Sobel filter to an edge to give a peak. This is then located to sub-pixel accuracy.

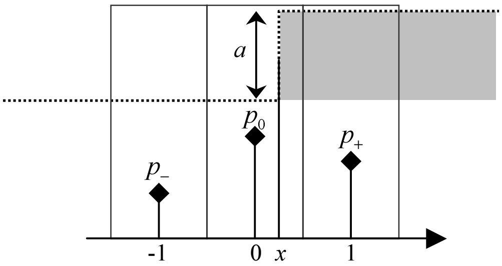

Figure 5: Applying a Sobel filter to an edge.

The pixel values associated with an edge of amplitude a are:

$$\begin{aligned} p_- &= a(\tfrac{1}{2} - x) \\ p_0 &= a \\ p_+ &= a(\tfrac{1}{2} + x) \end{aligned} \tag{19}$$

Solving for x gives

$$x = \frac{p_+ - p_-}{2p_0} \tag{20}$$

which is the same result as equation (15).

3 Effect of Using Each Model

In the previous section, several different models have been proposed. This section will examine the errors associated with using each model in a range of scenarios. Data used is artificial, to enable any systematic errors to be observed without the effects of noise. The models applied are:

1. Nearest pixel: the location of the local extremum.
2. Parabola: from equation (4).
3. Pyramid: from equation (8).
4. Centre of gravity: from equation (10) – COG.
5. Centre of gravity: from equation (12) – COG2.
6. Sobel edge: from equation (20).

3.1 Effect on Lines

Any object will become blurred when imaged because imaging sensors integrate the light within an area. The cross-section of a line can be considered as a rectangle function, which is then convolved by another rectangle function of width 1 pixel. Any lens blur present will also require a further convolution with the lens point spread function. In the examples examined here, the background will be set to 0 without loss of generality. Figure 6 shows the blurring effect of both imaging and lens blur (0.5 pixel Gaussian) on a 1.25 pixel wide line.

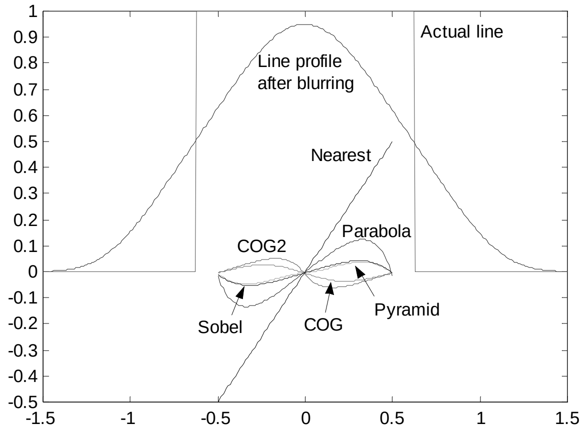

Figure 6: Profile of a 1.25 pixel wide line subject to 0.5 pixel Gaussian lens blur. Lower curves are error profiles of the different peak detection methods, with error measured as fractions of a pixel.

This profile is sampled for a range of sub-pixel offsets and the position of the local maximum estimated using the various models. The error between the actual and estimated offset is a systematic error introduced by the model.

The RMS error is calculated for the over the range of offsets to give an average systematic error in estimating the position of the line centre. This process was performed without lens blur (table 1), with 0.5 pixel lens blur (table 2) and 1 pixel lens blur (table 3).

Table 1: Lines with no lens blur. RMS error in line centre as a percentage of pixel width.

Method	**Line width (pixels)**							
	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
Nearest	29.2	29.2	29.2	29.2	29.2	29.2	29.2	29.2
Parabola	19.8	15.7	11.9	9.0	6.8	4.6	0	7.2
Pyramid	17.1	11.4	6.2	2.8	1.3	2.8	6.2	11.4
COG	14.4	7.2	0	4.3	4.8	3.1	0	4.1
COG2	14.4	7.2	0	4.6	6.8	9.0	11.9	15.7
Sobel	17.1	11.4	6.2	2.8	0.9	0.5	0	2.0

The nearest pixel method gives the same error regardless of the line width, and as expected, has the worst average error. All of the methods perform poorly for narrow line widths. The main reason for this is that there is insufficient data from the narrow lines to estimate the true line position. For lines of exactly 1 pixel width, both COG methods give an accurate location of the sub-pixel position of the line. For wider lines, equation (10) is better than equation (12) because the latter assumes (incorrectly) that the minimum of the 3 pixel values is the background. For narrow lines, both the Pyramid model and the Sobel model have identical error profiles because either p_- or p_+ is zero. However, without lens blur, the Pyramid model does not give a good fit to the actual profile. The fact that the Parabola model accurately estimates the offset for 2 pixel wide lines is a little surprising as that profile is trapezoidal rather than parabolic. Note though that the simpler Sobel method significantly outperforms the Parabola model without lens blur. The ideal response from the Sobel method is expected for 2 pixel wide lines from the analysis in section 2.4.

Table 1 may be considered the ideal situation, because there is no lens blur. In practise, there will always be some lens blur, and if the lens is well matched to the sensor, the lens point spread function is likely to be about 0.5 pixels wide. Therefore this case was also considered, with the results shown in table 2.

Table 2: Lines with 0.5 pixel Gaussian blur. RMS error in line centre as a percentage of pixel width.

Method	**Line width (pixels)**							
	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
Parabola	17.1	14.5	11.7	9.1	6.6	3.9	0.4	4.1
Pyramid	12.7	9.4	5.9	2.9	0.9	2.9	6.0	9.7
COG	7.7	3.7	0	2.2	2.5	1.6	2.0	5.3
COG2	7.7	3.6	0.6	3.9	6.6	9.1	11.8	14.8
Sobel	12.7	9.4	6.0	3.3	1.6	1.0	1.5	3.2

When lens blur is added, all of the models improved for lines less than 1 pixel wide. This is because the blur made the line wider, and therefore provided some more information to enable the offset to be estimated. For 1 pixel wide lines, again the COG method performs well. The added blur makes equation (12) less accurate however, because it is starting to violate the assumptions used in forming the model. The blur generally causes the Parabola and Pyramid methods to improve because the wider region of support makes the assumptions used there more valid. The Sobel method deteriorates with blur for larger line widths because the blur makes the assumptions illustrated in figure 3 less valid.

When the lens point spread function is 1 pixel wide, the image will start being noticeably blurred. This case is considered in table 3.

Table 3: Lines with 1 pixel Gaussian blur. RMS error in line centre as a percentage of pixel width.

Method	**Line width (pixels)**							
	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
Parabola	12.0	11.0	9.6	8.0	6.1	4.1	1.9	0.7
Pyramid	6.2	5.0	3.5	1.8	0.7	2.5	4.6	6.8
COG	1.3	1.0	0.8	1.3	2.1	3.7	5.8	8.5
COG2	0.5	1.6	3.1	4.9	6.8	8.7	10.6	12.4
Sobel	6.7	5.8	4.7	3.9	3.4	3.7	4.6	6.2

For narrow lines, the errors have decreased further, to the extent that the COG methods give reasonable results. For thick lines, the parabola method becomes increasingly effective as the profiles are becoming more parabolic in shape. The pyramid method is also increasingly accurate with more blur, and remains the best for lines about 1.5 pixels thick. The Sobel method deteriorates more significantly with thicker lines because the blur makes the model less accurate.

The analysis above implicitly assumes that the line is perpendicular to the three pixels used to locate the centre. However, in most practical situations, it cannot be guaranteed that lines will be exactly horizontal or vertical. If the line is at an angle, the rectangular pixel point spread function becomes trapezoidal, and therefore wider. Without loss of generality, assume that the line is at an angle of θ off the vertical. When this line is projected onto the horizontal scan direction, it results in a trapezoid, as shown in figure 7. This trapezoid can be considered as the convolution of two one-dimensional components: a vertical line of thickness $w/\cos\theta$ and a blur component of $\tan\theta$. The first component results in an apparent thickening of the line, and the second component can be considered as an increased blur in the direction of the horizontal pixels.

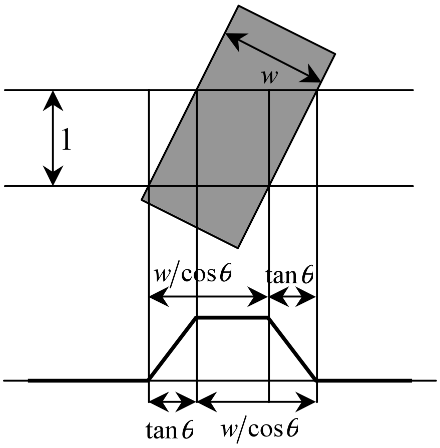

Figure 7: Projection of a diagonal line on a horizontal row of pixels.

The results of these two effects are outlined in the experiments described above. The above analysis assumed that the line was straight. If the line has significant curvature, the situation is significantly more complex.

3.2 Effect on edges

The effect on step edges was simulated by blurring the step edge by the rectangular pixel point spread function, and then a Gaussian lens point spread function. This is then passed through a Sobel filter to give the edge profile, as seen in figure 8.

This was applied to a step edge with a range of lens blurs, with the results listed in table 4. Clearly the simple COG does not work well, and neither does the Pyramid model. With no blur, the other 3 methods all gave perfect results. This was expected for the Sobel model, which is designed exactly for this situation, but the performance of the parabola model was a little surprising. Even more surprising is that the parabola method is significantly less sensitive to lens blur than the other methods.

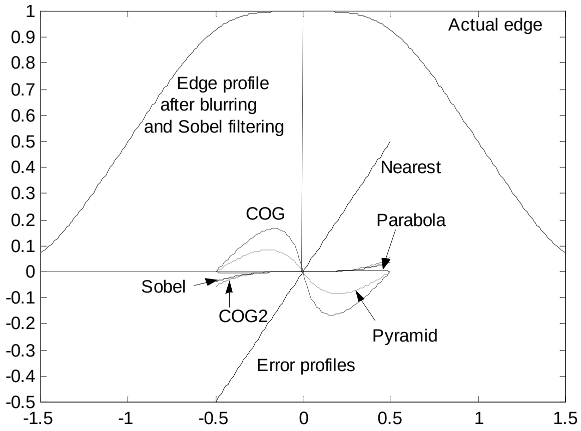

Figure 8: An edge and its profile with 0.5 Gaussian lens blur after Sobel filtering. Lower curves are error profiles of the different peak detection methods, with error measured as fractions of a pixel.

Table 4: Location of step edges after Sobel filtering. RMS error in position as a percentage of pixel width.

Method	**Width of lens PSF (pixels)**					
	0	0.25	0.5	0.75	1.0	1.25
Nearest	29.2	29.2	29.2	29.2	29.2	29.2
Parabola	0	0.1	0.4	1.1	1.9	2.6
Pyramid	6.2	6.2	6.0	5.3	4.6	3.9
COG	11.9	11.9	11.8	11.2	10.6	10.0
COG2	0	0.8	2.0	3.9	5.8	7.9
Sobel	0	0.5	1.5	3.0	4.6	6.3

This was applied to a step edge with a range of lens blurs, with the results listed in table 4. Clearly the simple COG does not work well, and neither does the Pyramid model. With no blur, the other 3 methods all gave perfect results. This was expected for the Sobel model, which is designed exactly for this situation, but the performance of the parabola model was a little surprising. Even more surprising is that the parabola method is significantly less sensitive to lens blur than the other methods.

3.3 Effect of Quantisation

Inevitably, whenever images are captured, the pixel values are quantised. The contrast between the object and background will govern the effective number of levels of quantisation, hence the quantisation noise introduced into the profile. In testing line location, the line thickness was chosen as that for which the method was designed and gives good results. The effect of quantisation on line location is summarised in table 5, with the result for edge location in table 6. In both cases, a lens blur of 0.5 pixels was used.

Table 5: Effect of quantisation on line location. RMS error in position as a percentage of pixel width.

Method	Line width	Levels				
		∞	256	64	32	16
Parobola	2.0	0.4	0.4	0.6	1.1	2.1
Pyramid	1.5	0.9	0.9	1.0	1.2	1.6
COG	1.0	0	0.2	0.6	1.1	2.1
COG2	1.0	0.6	0.6	0.8	1.1	2.0
Sobel	2.0	1.5	1.5	1.6	1.7	2.2

Table 6: Effect of quantisation on edge location. RMS error in position as a percentage of pixel width.

Method	Levels					
	∞	256	64	32	16	8
Parobola	0.4	0.4	0.6	1.1	2.1	4.4
COG2	2.0	2.0	2.1	2.2	2.5	4.7
Sobel	1.5	1.5	1.6	1.7	2.2	4.6

As expected, reducing the contrast increase the error in estimating the location of the lines or edges. The deterioration is most noticeable in situations where the accuracy is good to start with. When the contrast is low, there is little difference between the different methods.

3.4 Effect of Noise

The same exercise was repeated except adding zero-mean Gaussian noise. Lens blur was kept at 0.5 pixel widths, and no quantisation was applied.

Table 7: Effect of noise on line location. RMS error in position as a percentage of pixel width.

Method	Line width	SNR (dB)					
		∞	60	50	40	30	20
Parobola	2.0	0.4	0.4	0.5	1.1	3.4	13.6
Pyramid	1.5	0.9	0.9	0.9	1.2	2.7	8.5
COG	1.0	0	0.2	0.5	1.5	4.8	15.5
COG2	1.0	0.6	0.6	0.7	1.2	3.4	10.8
Sobel	2.0	1.5	1.5	1.5	1.7	2.9	8.1

When detecting lines, for typical signal to noise ratios found in images, there is very little to choose between the various methods (provided that an appropriate model is used). It is interesting to note that although the pyramid and Sobel models were not as good in low noise situations, they also tended to be less sensitive to noise.

Table 8: Effect of noise on edge location. RMS error in position as a percentage of pixel width.

Method	SNR (dB)					
	∞	60	50	40	30	20
Parobola	0.4	0.4	0.6	1.6	5.3	34.0
COG2	2.0	2.0	2.1	2.3	4.4	12.9
Sobel	1.5	1.5	1.6	1.9	4.3	13.3

The parabola method continued to excel for detecting edges in low noise situations. However, when the noise got heavier, there was a threshold effect, and the accuracy of the parabola method deteriorated rapidly.

4 Conclusions

When detecting lines and edges to sub-pixel accuracy, it is important to use an appropriate model. While any of the models discussed in this paper are better than no sub-pixel estimation, there are significant differences in the accuracy of the different models.

When detecting lines in images, the best line width to use is in the range 1 to 2 pixels. Narrower lines have a larger systematic error because their narrow extent does not always span more than 1 pixel. For 1 pixel wide lines, the centre of gravity gives the best estimate of true line location. For 2 pixel wide lines, a parabolic model gives the best fit.

When detecting step edges with a Sobel filter, a parabolic model provides the best estimate of the edge location. Other methods work well with no lens blur, but as some blur is always present, the parabolic method is least sensitive to this blur.

The accuracy of the estimation deteriorates with reducing contrast and increasing noise, as would be expected. For typical contrasts and signal to noise ratios, a sub-pixel accuracy of 1 – 2% of the width of a pixel can be expected.

5 Acknowledgements

The author would like to acknowledge financial support from Professor Sanjit Mitra, and the Signal and Image Processing Group at UCSB. This work was supported in part by a University of California MICRO grant with matching support from the Philips Research Laboratories, and in part by Microsoft Corp.

6 References

[1] Heikkila, J., Silven, O. "Calibration procedure for short focal length off-the-shelf CCD cameras." *Proceedings International Conference on Pattern Recognition*, 1: pp 166-170, (1996).

[2] Bailey, D.G., "A new approach to lens distortion correction", *Proceedings Image and Vision Computing New Zealand 2002*, pp 59-64 (2002).

[3] Weng J. Cohen P. Herniou M. "Camera Calibration with distortion models and accuracy evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(10): pp 965-980 (1992).

[4] Bailey D.G., Lill, T.H., "Image Registration Methods for Resolution Improvement", *Proceedings of Image and Vision Computing NZ*, pp 91-96 (August 1999).

[5] Jain, A.K., *Fundamentals of Image Processing.* Prentice Hall, Englewood Cliffs, New Jersey pp 400-407 (1989).

[6] Abdou, I.E., and Pratt, W.K., "Quantitative Design and Evaluation of Edge Enhancement / Thresholding Edge Detectors", *Proceedings of the IEEE* 67, pp 753-763 (1979).